Magnet Project API Documentation

Route Authentication & Role Protection Summary

Module	Route/Action	Public	Auth Required	Role(s) Required
Products	GET /api/products	Yes	No	None
	GET /api/products/:id	Yes	No	None (see below)
	POST /api/products/addProductsByBusiness	No	Yes	business
	POST /api/products/addProductsByMagnet_employee	No	Yes	magnet_employee, admin
	PUT /api/products/:id	No	Yes	business (own), admin, magnet_employee
	DELETE /api/products/:id	No	Yes	business (own), admin, magnet_employee
	PUT /api/products/:id/approve	No	Yes	admin, magnet_employee
	PUT /api/products/:id/decline	No	Yes	admin, magnet_employee
Categories	GET /api/categories	Yes	No	None
	POST /api/categories	No	Yes	business, admin, magnet_employee
	PUT /api/categories/:id	No	Yes	business (own), admin, magnet_employee
	DELETE /api/categories/:id	No	Yes	business (own), admin, magnet_employee
Orders	POST /api/orders	No	Yes	customer
	GET /api/orders/my	No	Yes	customer
	GET /api/orders/:id	No	Yes	admin, magnet_employee, customer (own)

Module	Route/Action	Public	Auth Required	Role(s) Required
	GET /api/orders	No	Yes	admin, magnet_employee
Reviews	POST /api/products/:id/reviews	No	Yes	customer
	GET /api/products/:id/reviews	Yes	No	None
	DELETE /api/reviews/:id	No	Yes	admin, magnet_employee
Wishlist	GET /api/wishlist	No	Yes	customer
	POST /api/wishlist	No	Yes	customer
	DELETE /api/wishlist/:productId	No	Yes	customer
Address	GET /api/addresses	No	Yes	customer
	POST /api/addresses	No	Yes	customer
	PUT /api/addresses/:id	No	Yes	customer
	DELETE /api/addresses/:id	No	Yes	customer
User	GET /api/user/profile	No	Yes	customer
	PUT /api/user/profile	No	Yes	customer
	GET /api/user/business-requests	No	Yes	admin, magnet_employee
	POST /api/user/business-approval	No	Yes	admin, magnet_employee
	GET /api/user/business/:businessId	No	Yes	admin, magnet_employee
	GET /api/user/business-profile	No	Yes	business, admin, magnet_employee
Auth	Registration/Login/OTP	Yes	No	None
	Forgot Password	No	Yes	Any authenticated user

Notes

- **Auth Required**: Requires a valid JWT token in the `Authorization` header.
- **Role(s) Required**: Requires the user to have one of the listed roles. If "(own)" is specified, the user must own the resource (e.g., their own product or order).
- **Public**: Anyone can access, no authentication required.
- **All**: All HTTP methods (GET, POST, PUT, DELETE) for that endpoint.

- **Business (if owns product in order)**: Business user can access if any product in the order is owned by them (see order tracking section).

Message Format

All API responses return bilingual messages in both English and Arabic for every endpoint, including all errors and successes. The system uses a centralized messages.js file to ensure consistency across all modules (auth, user, product, order, address, category, review, wishlist, etc.).

Example:

```
{
  "status": "success",
  "message": {
    "en": "User registered successfully",
    "ar": "تم تسجيل المستخدم بنجاح"
  },
  "data": { ... }
}
```

Verification Rules

Phone Number Verification

- **If phone number is provided**: Only the phone is verified (`isPhoneVerified = true`, `isEmailVerified = false`)
- **If no phone number is provided**: Only the email is verified (`isEmailVerified = true`, `isPhoneVerified = false`)

This applies to both customer and business registration.

Auth Routes

1. Register User

- **POST** `/api/auth/register`
- **Description:** Register a new customer user with automatic verification based on phone number presence.
- **Body:**
  - `firstname` (string, required)
  - `lastname` (string, required)
  - `email` (string, required)
  - `phone` (string, optional)
  - `password` (string, required)
  - `country` (string, required)
  - `language` (string, optional, default: 'en')

- **Verification Logic:**
  - If `phone` is provided: `isPhoneVerified = true`, `isEmailVerified = false`
  - If `phone` is not provided: `isEmailVerified = true`, `isPhoneVerified = false`
- **Response:**
  - `201 Created`: User info + JWT token with verification status

---

2. Register Business

- **POST** `/api/auth/business-register`
- **Description:** Register a new business user (requires approval) with automatic verification based on phone number presence.
- **Body:**
  - `firstname`, `lastname`, `email`, `password`, `crNumber`, `vatNumber`, `companyName`, `companyType`, `country`, `city`, `district`, `streetName`, `phone` (all required)
- **Verification Logic:**
  - If `phone` is provided: `isPhoneVerified = true`, `isEmailVerified = false`
  - If `phone` is not provided: `isEmailVerified = true`, `isPhoneVerified = false`
- **Response:**
  - `201 Created`: Business info (under review) with verification status

---

3. Send Email OTP

- **POST** `/api/auth/send-email-otp`
- **Description:** Send an OTP to a new email (fails if email already exists).
- **Body:**
  - `email` (string, required)
- **Response:**
  - `200 OK`: Success message (bilingual)

---

4. Send Phone OTP

- **POST** `/api/auth/send-phone-otp`
- **Description:** Send an OTP to a new phone (fails if phone already exists).
- **Body:**
  - `phone` (string, required)
- **Response:**
  - `200 OK`: Success message (bilingual)

---

5. Confirm OTP

- **POST** `/api/auth/confirm-otp`
- **Description:** Confirm an OTP for any email or phone identifier (works for both registered and non-registered users).
- **Body:**
  - `identifier` (string, required)

    - `otp` (string, required)
- **Response:**
    - `200 OK`: OTP verified successfully (bilingual message)

---

6. Confirm Login OTP

- **POST** `/api/auth/confirm-login-otp`
- **Description:** Confirm an OTP for registered users and return user data with JWT token (like login API).
- **Body:**
    - `identifier` (string, required)
    - `otp` (string, required)
- **Response:**
    - `200 OK`: User info + JWT token (bilingual message)
    - `404 Not Found`: User not found (bilingual message)

---

7. Login

- **POST** `/api/auth/login`
- **Description:** Login with email/phone and password.
- **Body:**
    - `identifier` (email or phone, required)
    - `password` (string, required)
- **Response:**
    - `200 OK`: User info + JWT token (bilingual message)

---

8. Login with OTP

- **POST** `/api/auth/login-with-otp`
- **Description:** Request an OTP for login (email or phone).
- **Body:**
    - `identifier` (email or phone, required)
- **Response:**
    - `200 OK`: OTP sent (bilingual message)

---

9. Forgot Password

- **POST** `/api/auth/forgot-password`
- **Description:** Change password for authenticated user.
- **Headers:** `Authorization: Bearer <token>`
- **Body:**
    - `newPassword` (string, required)
- **Response:**
    - `200 OK`: Password updated (bilingual message)

---

10. Create Admin User

- **POST** `/api/auth/create-admin`
- **Description:** Create a new admin user. Only accessible by authenticated admin users.
- **Headers:** `Authorization: Bearer <token>`
- **Body:**
  - `firstname` (string, required)
  - `lastname` (string, required)
  - `email` (string, required)
  - `phone` (string, optional)
  - `password` (string, required)
  - `country` (string, required)
  - `language` (string, optional, default: 'en')
- **Response:**
  - `201 Created`: Admin user info (bilingual message)
  - `403 Forbidden`: Insufficient permissions (bilingual message)
  - `400 Bad Request`: Email/phone already registered (bilingual message)

---

11. Create Magnet Employee User

- **POST** `/api/auth/create-magnet-employee`
- **Description:** Create a new magnet employee user. Only accessible by authenticated admin users.
- **Headers:** `Authorization: Bearer <token>`
- **Body:**
  - `firstname` (string, required)
  - `lastname` (string, required)
  - `email` (string, required)
  - `phone` (string, optional)
  - `password` (string, required)
  - `country` (string, required)
  - `language` (string, optional, default: 'en')
- **Response:**
  - `201 Created`: Magnet employee user info (bilingual message)
  - `403 Forbidden`: Insufficient permissions (bilingual message)
  - `400 Bad Request`: Email/phone already registered (bilingual message)

Product Routes

1. Get Products

- **GET** `/api/products`
- **Description:** Get all approved products (public, any user can call). Admin/magnet_employee can see all products.
- **Response:**
  - `200 OK`: List of products
- **Product Object:**

  - `code` (string, auto-generated if not provided, e.g. "A001")
  - `category` (string, required)
  - `name` (string, required)
  - `images` (array of strings)
  - `description` (string)
  - `attachments` (array of product IDs, references to other products)
  - `unit` (string)
  - `minOrder` (number)
  - `pricePerUnit` (string)
  - `stock` (number)
  - `customFields` (array of objects, min 5, max 10, each: `{ key, value }`)
  - `status` (string: 'pending', 'approved', 'declined')
  - `owner` (user id, required)
  - `approvedBy` (user id, admin/employee who approved)
  - `createdAt` (date)

1a. Get Product by ID

- **GET** `/api/products/:id`
- **Description:** Get a single product by its ID. Publicly accessible for approved products. Admin, business, and magnet_employee can access any product (including pending/declined).
- **Response:**
  - `200 OK`: Product object (see above)
  - `404 Not Found`: If product does not exist
  - `403 Forbidden`: If trying to access a non-approved product without proper role

---

2. Add Product (Business)

- **POST** `/api/products/addProductsByBusiness`
- **Description:** Business user adds a new product (pending approval). Product code is auto-generated if not provided.
- **Headers:** `Authorization: Bearer <token>`
- **Body:**
  - `category` (string, required)
  - `name` (string, required)
  - `images` (array of strings, optional)
  - `attachments` (array of product IDs, references to other products, optional)
  - `description`, `unit`, `minOrder`, `pricePerUnit`, `stock`, `customFields` (all optional)
  - `customFields` (array of objects, min 5, max 10, each: `{ key, value }`, required)
- **Response:**
  - `201 Created`: Product info (pending approval) with bilingual message

---

3. Add Product (Magnet Employee)

- **POST** `/api/products/addProductsByMagnet_employee`

- **Description:** Magnet employee adds a new product (approved immediately). Product code is auto-generated if not provided.
- **Headers:** `Authorization: Bearer <token>`
- **Body:**
  - `category` (string, required)
  - `name` (string, required)
  - `images` (array of strings, optional)
  - `attachments` (array of product IDs, references to other products, optional)
  - `description`, `unit`, `minOrder`, `pricePerUnit`, `stock`, `customFields` (all optional)
  - `customFields` (array of objects, min 5, max 10, each: `{ key, value }`, required)
  - `owner` (user id, required)
- **Response:**
  - `201 Created`: Product info (approved) with bilingual message

---

4. Update Product

- **PUT** `/api/products/:id`
- **Description:** Business user updates their own product (pending approval). Admin/magnet_employee can update and approve/decline.
- **Headers:** `Authorization: Bearer <token>`
- **Body:**
  - Any product field (see above)
  - `status` (optional, only admin/magnet_employee can set to 'approved' or 'declined')
- **Response:**
  - `200 OK`: Updated product info with bilingual message

---

5. Delete Product

- **DELETE** `/api/products/:id`
- **Description:** Business user deletes their own product, or admin/employee deletes any product.
- **Headers:** `Authorization: Bearer <token>`
- **Response:**
  - `200 OK`: Product deleted with bilingual message

---

6. Approve Product

- **PUT** `/api/products/:id/approve`
- **Description:** Admin/magnet_employee approves a product.
- **Headers:** `Authorization: Bearer <token>`
- **Response:**
  - `200 OK`: Product approved with bilingual message

---

7. Decline Product

- **PUT** `/api/products/:id/decline`

- **Description:** Admin/magnet_employee declines a product.
- **Headers:** `Authorization: Bearer <token>`
- **Response:**
  - `200 OK`: Product declined with bilingual message

Real-Time Order Tracking (WebSocket)

Overview

- The API supports real-time order status updates using WebSocket (Socket.IO).
- Clients can subscribe to updates for specific orders and receive instant notifications when the order status changes (e.g., confirmed, shipped, delivered).

How It Works

1. **Client authenticates with a JWT token when connecting to the WebSocket.**
2. **Client joins a room for a specific order using the order ID.**
3. **When the order status changes, the server emits an `orderStatusUpdate` event to all clients in that room.**

WebSocket Connection (Socket.IO)

- **URL:** `ws://<your-server>:5000` (or `wss://` for HTTPS)
- **Library:** socket.io-client

Client Example (JavaScript/React):

```
import { io } from 'socket.io-client';
const socket = io('http://localhost:5000', {
  auth: { token: '<JWT_TOKEN>' }
});

// Join the order room
socket.emit('joinOrderRoom', '<ORDER_ID>');

// Listen for updates
socket.on('orderStatusUpdate', (data) => {
  // data: { orderId, status, statusLog, updatedAt }
  console.log('Order update:', data);
});
```

Order Room Access Rules

- **Admin & magnet_employee:** Can track any order.
- **Business:** Can track orders that include products they own.
- **Customer:** Can track only their own orders.
- Unauthorized attempts to join order rooms will be ignored.

Server-Side Security

- The server authenticates each socket connection using the JWT token.
- The server enforces the above access rules for joining order rooms.

Order Status Update Event

- **Event:** `orderStatusUpdate`
- **Payload:**

```
{
  "orderId": "...",
  "status": "shipped",
  "statusLog": [
    { "status": "pending", "timestamp": "..." },
    { "status": "confirmed", "timestamp": "..." },
    { "status": "shipped", "timestamp": "..." }
  ],
  "updatedAt": "2024-07-10T12:00:00.000Z"
}
```

Security Notes

- Clients must provide a valid JWT token when connecting.
- The server will only allow joining order rooms for orders the user is authorized to track (see access rules above).
- Unauthorized attempts to join rooms will be ignored.

Category Routes

1. Get Categories

- **GET** `/api/categories`
- **Description:** Get all categories (public).
- **Response:**
  - `200 OK`: List of categories

2. Create Category

- **POST** `/api/categories`
- **Description:** Create a new category (business, admin, or magnet_employee only).
- **Headers:** `Authorization: Bearer <token>`
- **Body:**
  - `name` (string, required)
  - `description` (string, optional)
- **Response:**
  - `201 Created`: Category info (bilingual message)

3. Update Category

- **PUT** `/api/categories/:id`
- **Description:** Update a category (business, admin, or magnet_employee only).
- **Headers:** `Authorization: Bearer <token>`
- **Body:**
  - `name` (string, optional)
  - `description` (string, optional)
- **Response:**
  - `200 OK`: Updated category info (bilingual message)

4. Delete Category

- **DELETE** `/api/categories/:id`
- **Description:** Delete a category (business, admin, or magnet_employee only).
- **Headers:** `Authorization: Bearer <token>`
- **Response:**
  - `200 OK`: Category deleted (bilingual message)

Order Routes

1. Create Order

- **POST** `/api/orders`
- **Description:** Create a new order (customer only).
- **Headers:** `Authorization: Bearer <token>`
- **Body:**
  - Order details (see order model)
- **Response:**
  - `201 Created`: Order info (bilingual message)

2. Get My Orders

- **GET** `/api/orders/my`
- **Description:** Get all orders for the authenticated customer.
- **Headers:** `Authorization: Bearer <token>`
- **Response:**
  - `200 OK`: List of orders (bilingual message)

3. Get Order by ID

- **GET** `/api/orders/:id`
- **Description:** Get order by ID (admin, magnet_employee, or owner).
- **Headers:** `Authorization: Bearer <token>`
- **Response:**
  - `200 OK`: Order info (bilingual message)

4. Get All Orders

- **GET** `/api/orders`
- **Description:** Get all orders (admin, magnet_employee only).
- **Headers:** `Authorization: Bearer <token>`
- **Response:**
  - `200 OK`: List of orders (bilingual message)

5. Update Order Status

- **PUT** `/api/orders/:id/status`
- **Description:** Update order status (admin, magnet_employee only).
- **Headers:** `Authorization: Bearer <token>`
- **Body:**
  - `status` (string, required)
- **Response:**
  - `200 OK`: Updated order info (bilingual message)

Address Routes

1. Get Addresses

- **GET** `/api/addresses`
- **Description:** Get all addresses for the authenticated customer.
- **Headers:** `Authorization: Bearer <token>`
- **Response:**
  - `200 OK`: List of addresses (bilingual message)

2. Add Address

- **POST** `/api/addresses`
- **Description:** Add a new address (customer only).
- **Headers:** `Authorization: Bearer <token>`
- **Body:**
  - `addressLine1` (string, required)
  - `addressLine2` (string, optional)
  - `city` (string, required)
  - `state` (string, required)
  - `postalCode` (string, required)
  - `country` (string, required)
- **Response:**
  - `201 Created`: Address info (bilingual message)

3. Update Address

- **PUT** `/api/addresses/:id`
- **Description:** Update an address (customer only).
- **Headers:** `Authorization: Bearer <token>`
- **Body:**
  - Any address field (see above)

- **Response:**
  - `200 OK`: Updated address info (bilingual message)

4. Delete Address

- **DELETE** `/api/addresses/:id`
- **Description:** Delete an address (customer only).
- **Headers:** `Authorization: Bearer <token>`
- **Response:**
  - `200 OK`: Address deleted (bilingual message)

Wishlist Routes

1. Get Wishlist

- **GET** `/api/wishlist`
- **Description:** Get the authenticated customer's wishlist.
- **Headers:** `Authorization: Bearer <token>`
- **Response:**
  - `200 OK`: List of wishlist items (bilingual message)

2. Add to Wishlist

- **POST** `/api/wishlist`
- **Description:** Add a product to the authenticated customer's wishlist.
- **Headers:** `Authorization: Bearer <token>`
- **Body:**
  - `productId` (string, required)
- **Response:**
  - `200 OK`: Wishlist item info (bilingual message)

3. Remove from Wishlist

- **DELETE** `/api/wishlist/:productId`
- **Description:** Remove a product from the authenticated customer's wishlist.
- **Headers:** `Authorization: Bearer <token>`
- **Response:**
  - `200 OK`: Wishlist item removed (bilingual message)

Review Routes

1. Add Review

- **POST** `/api/products/:id/reviews`
- **Description:** Add a review to a product (customer only).
- **Headers:** `Authorization: Bearer <token>`
- **Body:**
  - `rating` (number, required)

    - `comment` (string, optional)
- **Response:**
    - `201 Created`: Review info (bilingual message)

2. Get Product Reviews

- **GET** `/api/products/:id/reviews`
- **Description:** Get all reviews for a product (public).
- **Response:**
    - `200 OK`: List of reviews (bilingual message)

3. Delete Review

- **DELETE** `/api/reviews/:id`
- **Description:** Delete a review (admin or magnet_employee only).
- **Headers:** `Authorization: Bearer <token>`
- **Response:**
    - `200 OK`: Review deleted (bilingual message)

User Routes

1. Get User Profile

- **GET** `/api/user/profile`